Session 509

Java Security on Mac OS X

Scott Kovatch
Engineer, Java Runtime Classes

Introduction

- We want to answer your questions about Java security on Mac OS X
  - Is there any difference in Apple's implementation of Java security and Sun's implementation on Windows and Solaris?
  - Will my signed applet work on Mac OS X?
  - How do you make a secure connection without Internet Explorer?

What You'll Learn

- Introduction to Java2's security model, plus Mac OS X-specific details
- How to sign JAR files for use with Internet Explorer for Mac OS X and Windows and Netscape
- How to make secure connections without Internet Explorer
- Tips and tricks for debugging security-related problems in your code

First, Some News

- Security architecture in Java on Mac OS X is *completely* identical to other platforms
- No support for extra features in Applet Launcher or command-line appletviewer
- Signed applet support is the same as that in MRJ

Where We've Been

JDK 1.1.X Security Model

- All or nothing proposition
  - Sandbox restrictions on applets too limiting
  - Security checks not easily extensible
  - Locally installed applets too easily trusted
  - Security managers are too easily broken by changes in the code being protected

Where We're Going

Java2 Security Model

- Policy-based architecture
- Policies are collections of permissions specified for particular codebases
- Code is granted permission to perform certain activities based on where it came from
- No code is trusted by default, regardless of location

Policy File Entries

```
grant signedBy “signer-names”, codeBase “URL” {
        permission <permission-class-name>
                “target-name”, “action”,
                signedBy “signer-names”
        ...
};
```

Establishing the Policy

- Security policies can be set in three ways
  - System-wide:

    ${JAVA_HOME}/lib/security/java.policy
  - Per user:

    ~/.java.policy
  - Per application: Bundle a policy specification inside your Java application
- Policy files do not supercede each other

An Example

- Wrap JINI as an application
  - We'll make two applications, one to wrap the basic JINI service providers and one for the simple service browser
  - The browser needs its own policy file for the additional permissions in JINI
- We can do it all with MRJAppBuilder

Why Bother?

- Policy files make sense when you control the environment
  - In-house applications
  - Special-purpose networking
  - Intranet applets
- But what if you can't control it?
- . . . Then signed JAR files are for you

Signed JAR Files on Mac OS X

- Available in Java for Mac OS X embedded in Internet Explorer
  - Provides pop-up dialogs to let users validate signers
  - Supports JARs signed for MRJ 2.2.x
- With some extra set up work it also
  - Supports RSA-signed applets
  - Supports standard Verisign and Thawte RSA certificates
  - Provides identical RSA signed applet support as Internet Explorer and Netscape Navigator

Signed JAR Files on Mac OS X

- What does ‘some extra work’ mean?
  - Install Java Secure Sockets Extension 1.0.2
  - Copy over sunrsasign.jar from a Windows distribution

Signed JAR Files on Mac OS X

- Signed JAR files get full execution permissions
  - Behave the same way as signed JARs in MRJ
  - Equivalent to java.security.AllPermission
- Can use either Java2's code signing tools or Netscape's signing tools
- Currently no Java "control panel" for management of certificates

Demo

Ted Jucevic
DTS Engineer

Debugging Security Problems

- You can find out why you're getting a security exception
- Set the property java.security.debug
  - **java -Djava.security.debug=help** shows you all available options
  - Start with **-Djava.security.debug=access** to see what permissions are being tested
  - Add 'stack' or 'failure' to get more information

Debugging Security Problems

- Use the security features to help you test a signed applet—before it's signed!
- Create a file called **.java.policy** in your home directory with this entry:

```
grant codeBase http://my.server.com/applet.jar {
        permission java.security.AllPermission;
}
```

Secure Connections in MRJ

- MRJ only supported secure connections of the form

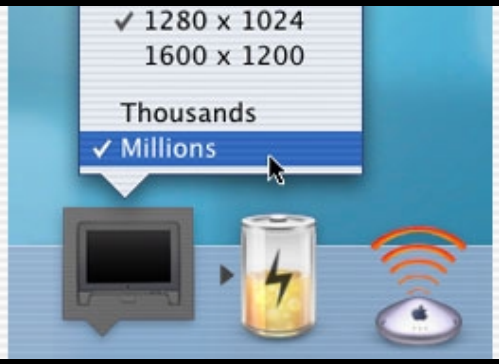
```
URL myURL = new
  URL("https://www.amazon.com");
URLConnection uc =
  myURL.getConnection();
```

- Even then, only inside of Internet Explorer 5.0

SSL Connections in Mac OS X

- Java Secure Sockets Extension (JSSE)
  - A pure Java implementation of SSL v3 and TLS
  - See **http://java.sun.com/products/jsse** for downloading and installation
  - You can bundle it in your application
  - 'Non-commercial reference implementation'
  - Watch what you download: US/Canada version allows third-party plug-in providers

Demo



SecureFTP

What Can You Do With JSSE?

- Create an https-based URLConnection
  - Almost identical to what you do now
- Create a secure connection at the Socket API level
- Once you have a socket or URLConnection you talk to a secure connection the same way as a plain connection

How Do You Use JSSE?

- Three JARs you can bundle into your application
- Your code needs to register the SSL provider via

```
java.security.Security.addProvider
        (new com.sun.net.ssl.internal.ssl.Provider());
```

- What you do next depends on your needs

HTTPS Connections

- Must define the property

  java.protocol.handler.pkgs = com.sun.net.ssl.internal.www.protocol

- You can do this from the command line or set it in your application's property file with MRJAppBuilder, or set it explicitly with setProperty

Secure Sockets

- Very similar to a plain socket, but with a few extra steps
  - Get an instance of the SSLSocketFactory
  - Ask the factory to create a SSLSocket to a host
  - Open an input or output stream on the socket

JSSE Resources

JSSE home page
http://java.sun.com/products/jsse

Dr. Dobbs article on using JSSE
http://www.ddj.com/articles/2001/0102/0102a/0102a.htm

What's Next?

- Today's session was just an introduction
  - Pick up a book on Java security
- Install JSSE
- . . . But first, go to the Java Feedback Forum in Room J1!

Who to Contact

Alan Samuel
Java Technology Manager
Apple Worldwide Developer Relations
bluker1@apple.com

Q&A

http://developer.apple.com/events/urls.html

WWDC
Worldwide Developers Conference 2001